

LPORAN JOBSHEET 5

SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

Mata Kuliah : Algoritma dan Struktur Data

Dosen : **Mungki Astiningrum, S.T., M.Kom.**



Ilham Dharma Atmaja

244107020220

Kelas : 1A

Absen : 14

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025**

5.1 Praktikum 1 - Mengimplementasikan Sorting menggunakan object

5.2.1. Hasil Algoritma (searching bubble sort, selection sort dan insertion sort)

```
package Praktikum05;

public class Sorting14 {

    int[] data;
    int jumData;

    Sorting14(int Data[], int jumData) {
        this.jumData = jumData;
        data = new int[jumData];
        for (int i = 0; i < jumData; i++) {
            data[i] = Data[i];
        }
    }

    void bubbeleSort() {
        int temp = 0;
        for (int i = 0; i < jumData - 1; i++) {
            for (int j = 0; j < jumData - 1; j++) {
                if (data[j] > data[j + 1]) {
                    temp = data[j];
                    data[j] = data[j + 1];
                    data[j + 1] = temp;
                }
            }
        }
    }

    void tampil() {
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i] + " ");
        }
        System.out.println();
    }
}
```

```

void SelectionSort() {
    for (int i=0; i<jumData-1; i++) {
        int min = i;
        for (int j=i+1; j<jumData; j++) {
            if (data[j] < data[min]) {
                min = j;
            }
        }
        int temp = data[min];
        data[min] = data[i];
        data[i] = temp;
    }
}

void insertionSort() {
    for (int i=1; i<jumData; i++) {
        int temp = data[i];
        int j = i;
        while (j>0 && data[j-1] > temp) {
            data[j] = data[j-1];
            j--;
        }
        data[j] = temp;
    }
}

class MainSorting14 {

    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        Sorting14 dataurut1 = new Sorting14(a, a.length);

        System.out.println("Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbeleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut1.tampil();
    }
}

```

```

        int b[] = {30, 20, 2, 8, 14};
        Sorting14 dataurut2 = new Sorting14(b, b.length);
        System.out.println("Data awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut2.tampil();

        int c[] = {40, 30, 2, 9, 16};
        Sorting14 dataurut3 = new Sorting14(c, c.length);
        System.out.println("Data awal 3");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut3.tampil();
    }
}
}

```

5.2.2 Verifikasi Hasil Percobaan

a. searching bubble sort

```

Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
PS C:\Users\ILHAM DHARMA A\OneDrive\Documents\SEMESTER

```

b. selection sort

```

Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 8 14 20 30

```

c. insertion sort

```

Data awal 3
40 30 2 9 16
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 9 16 30 40

```

5.2.3. Pertanyaan

1. Fungsi dari kode tersebut untuk menukar posisi dua elemen dalam array contohnya apabila elemen sebelumnya `data[j-1]` lebih besar dari elemen sekarang `data[j]`

2.

```
for (int i=0; i<jumData-1; i++) {  
    int min = i;  
    for (int j=i+1; j<jumData; j++) {  
        if (data[j] < data[min]) {  
            min = j;  
        }  
    }  
    int temp = data[min];  
    data[min] = data[i];  
    data[i] = temp;  
}
```

3. pada Insertionsort ini menjelaskan bahwa kondisi ini digunakan untuk memindahkan elemen yang lebih besar dari `temp` ke satu posisi ke kanan. Ini dilakukan untuk membuat ruang bagi elemen `temp` agar dapat ditempatkan pada posisi yang benar
4. perintah `data[j+1] = data[j]` digunakan untuk memindahkan elemen `data[j]` ke satu posisi ke kanan agar dapat membuat ruangan bagi elemen `temp`

5.2 Praktikum 2- (Sorting Menggunakan Array of Object)

5.2.1 Langkah Praktikum - Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

```
package Praktikum05;

public class Mahasiswa14 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa14(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi() {
        System.out.println("NIM : " + nim);
        System.out.println("Nama : " + nama);
        System.out.println("Kelas : " + kelas);
        System.out.println("IPK : " + ipk);
    }
}

class MahasiswaBerprestasi {
    Mahasiswa14 [] listMhs= new Mahasiswa14[5];
    int idx;

    void tambah(Mahasiswa14 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil () {
        for (Mahasiswa14 m : listMhs) {
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    void bubbleSort() {
        for (int i=0; i<listMhs.length-1; i++) {
            for (int j=1; j<listMhs.length-i; j++) {
                if (listMhs[j].ipk > listMhs[j-1].ipk) {
                    Mahasiswa14 temp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = temp;
                }
            }
        }
    }
}

class MahasiswaDemo14 {
    public static void main(String[] args) {
        Mahasiswa14 m1 = new Mahasiswa14("123", "Zidan", "2A", 3.2);
        Mahasiswa14 m2 = new Mahasiswa14("124", "Ayu", "2A", 3.5);
    }
}
```

```

        Mahasiswa14 m3 = new Mahasiswa14("125", "Sofi", "2A", 3.1);
        Mahasiswa14 m4 = new Mahasiswa14("126", "Sita", "2A", 3.9);
        Mahasiswa14 m5 = new Mahasiswa14("127", "Miki", "2A", 3.7);

        MahasiswaBerprestasi List = new MahasiswaBerprestasi();
        List.tambah(m1);
        List.tambah(m2);
        List.tambah(m3);
        List.tambah(m4);
        List.tambah(m5);

        System.out.println("Data Mahasiswa Sebelum sorting: ");
        List.tampil();
        System.out.println("Data Mahasiswa Setelah sorting berdasarkan IPK
(DESC) :");
        List.bubbleSort();
        List.tampil();
    }
}

```

5.2.2 Verifikasi Hasil percobaan

```

Data Mahasiswa Sebelum sorting:
NIM : 123
Nama : Zidan
Kelas : 2A
IPK : 3.2
-----
NIM : 124
Nama : Ayu
Kelas : 2A
IPK : 3.5
-----
NIM : 125
Nama : Sofi
Kelas : 2A
IPK : 3.1
-----
NIM : 126
Nama : Sita
Kelas : 2A
IPK : 3.9
-----
NIM : 127
Nama : Miki
Kelas : 2A
IPK : 3.7
-----
Data Mahasiswa Setelah sorting berdasarkan IPK (DESC) :
NIM : 126
Nama : Sita
Kelas : 2A
IPK : 3.9
-----
NIM : 127
Nama : Miki
Kelas : 2A
IPK : 3.7
-----
NIM : 124
Nama : Ayu
Kelas : 2A
IPK : 3.5
-----
NIM : 123
Nama : Zidan
Kelas : 2A
IPK : 3.2
-----
NIM : 125
Nama : Sofi
Kelas : 2A
IPK : 3.1
-----

```

5.2.3 Jawaban Pertanyaan

1.
 - a. Syarat `i < listMhs.length` digunakan karena pada iterasi terakhir, elemen terbesar sudah berada di posisi yang benar jadi tidak perlu dibandingkan lagi
 - b. Syarat `j < listMhs.length - i` digunakan karena pada setiap iterasi `i`, elemen terbesar dari sub array yang belum diurutkan akan menyesuaikan ke posisi yang benar
 - c. Jika ada banyak data pada `listMhs` adalah 50 maka perulangan `i` akan berlangsung banyak $50 - 1 = 49$ kali setiap tahap double sort akan menggelembungkan elemen terbesar dari sub array yang belum diurutkan ke posisi yang benar

2. `package Praktikum05;`

```
import java.util.Scanner;

class Mahasiswa14 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    Mahasiswa14(String nim, String nama, String kelas, double ipk) {

        this.nim = nim;

        this.nama = nama;

        this.kelas = kelas;

        this.ipk = ipk;

    }

    void tampilInformasi() {

        System.out.println("NIM : " + nim);

        System.out.println("Nama : " + nama);

        System.out.println("Kelas : " + kelas);

        System.out.println("IPK : " + ipk);

    }

}

class MahasiswaBerprestasi {

    Mahasiswa14 [] listMhs;

    int idx;

    MahasiswaBerprestasi(int jumlahMahasiswa) {

        listMhs = new Mahasiswa14[jumlahMahasiswa];

    }

}
```



```

    }

    void tambah(Mahasiswa14 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil () {
        for (Mahasiswa14 m : listMhs) {
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    void bubbleSort() {
        for (int i=0; i<listMhs.length-1; i++) {
            for (int j=1; j<listMhs.length-i; j++) {
                if (listMhs[j].ipk > listMhs[j-1].ipk) {
                    Mahasiswa14 temp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = temp;
                }
            }
        }
    }
}

class MahasiswaDemo14 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Masukkan jumlah mahasiswa: ");
        int jumlahMahasiswa = scanner.nextInt();
        scanner.nextLine();

        MahasiswaBerprestasi List = new MahasiswaBerprestasi(jumlahMahasiswa);
    }
}

```

```

        for (int i = 0; i < jumlahMahasiswa; i++) {
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));
            System.out.print("NIM: ");
            String nim = scanner.nextLine();
            System.out.print("Nama: ");
            String nama = scanner.nextLine();
            System.out.print("Kelas: ");
            String kelas = scanner.nextLine();
            System.out.print("IPK: ");
            double ipk = scanner.nextDouble();
            scanner.nextLine();

            Mahasiswa14 mahasiswa = new Mahasiswa14(nim, nama, kelas, ipk);
            List.tambah(mahasiswa);
        }

        System.out.println("Data Mahasiswa Sebelum sorting: ");
        List.tampil();

        System.out.println("Data Mahasiswa Setelah sorting berdasarkan IPK
(DESC) :");
        List.bubbleSort();
        List.tampil();
    }
}

```

```

Masukkan jumlah mahasiswa: 2
Masukkan data mahasiswa ke-1
NIM: 123
Nama: ian
Kelas: 2C
IPK: 3,4
Masukkan data mahasiswa ke-2
NIM: 124
Nama: yuyull
Kelas: 2C
IPK: 3,8
Data Mahasiswa Sebelum sorting:
NIM : 123
Nama : ian
Kelas : 2C
IPK : 3.4
-----
NIM : 124
Nama : yuyull
Kelas : 2C
IPK : 3.8
-----
Data Mahasiswa Setelah sorting berdasarkan IPK (DESC) :
NIM : 124
Nama : yuyull
Kelas : 2C
IPK : 3.8
-----
NIM : 123
Nama : ian
Kelas : 2C
IPK : 3.4
-----

```

Hasil Percobaan

5.3.4 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

```
package Praktikum05;

public class Mahasiswa14 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    Mahasiswa14(String nim, String nama, String kelas, double ipk) {

        this.nim = nim;

        this.nama = nama;

        this.kelas = kelas;

        this.ipk = ipk;

    }

    void tampilInformasi() {

        System.out.println("NIM : " + nim);

        System.out.println("Nama : " + nama);

        System.out.println("Kelas : " + kelas);

        System.out.println("IPK : " + ipk);

    }

}

class MahasiswaBerprestasi {

    Mahasiswa14 [] listMhs= new Mahasiswa14[5];

    int idx;

    void tambah(Mahasiswa14 m) {

        if (idx < listMhs.length) {

            listMhs[idx] = m;

            idx++;

        } else {

            System.out.println("Data sudah penuh");

        }

    }

}
```

```

    }

    void tampil () {
        for (Mahasiswa14 m : listMhs) {
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    void bubbleSort() {
        for (int i=0; i<listMhs.length-1; i++) {
            for (int j=1; j<listMhs.length-i; j++) {
                if (listMhs[j].ipk > listMhs[j-1].ipk) {
                    Mahasiswa14 temp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = temp;
                }
            }
        }
    }

    void SelectionSort() {
        for (int i=0; i<listMhs.length-1; i++) {
            int idxmin = i;
            for (int j=i+1; j<listMhs.length; j++) {
                if (listMhs[j].ipk < listMhs[idxmin].ipk) {
                    idxmin = j;
                }
            }
            Mahasiswa14 temp = listMhs[idxmin];
            listMhs[idxmin] = listMhs[i];
            listMhs[i] = temp;
        }
    }
}

class MahasiswaDemo14 {
    public static void main(String[] args) {
        Mahasiswa14 m1 = new Mahasiswa14("123", "Ali", "2A", 3.9);
        Mahasiswa14 m2 = new Mahasiswa14("124", "Ila", "2A", 3.1);
    }
}

```

```

Mahasiswa14 m3 = new Mahasiswa14("125", "Agus", "2A", 3.6);
Mahasiswa14 m4 = new Mahasiswa14("126", "Tika", "2A", 3.3);
Mahasiswa14 m5 = new Mahasiswa14("127", "Udin", "2A", 3.2);

MahasiswaBerprestasi List = new MahasiswaBerprestasi();

List.tambah(m1);
List.tambah(m2);
List.tambah(m3);
List.tambah(m4);
List.tambah(m5);

System.out.println("Data Mahasiswa Sebelum sorting: ");
List.tampil();

System.out.println("Data Mahasiswa Setelah sorting berdasarkan IPK (DESC)
:");

List.bubbleSort();
List.tampil();

System.out.println("Data yang sudah teurut menggunakan SELECTION SORT (ASC)
:");

List.SelectionSort();
List.tampil();
}
}

```

Hasil Output

```
Data Mahasiswa Sebelum sorting:
NIM : 123
Nama : Ali
Kelas : 2A
IPK : 3.9
-----
NIM : 124
Nama : Ila
Kelas : 2A
IPK : 3.1
-----
NIM : 125
Nama : Agus
Kelas : 2A
IPK : 3.6
-----
NIM : 126
Nama : Tika
Kelas : 2A
IPK : 3.3
-----
NIM : 127
Nama : Udin
Kelas : 2A
IPK : 3.2
-----
Data yang sudah teurut menggunakan SELECTION SORT (ASC) :
NIM : 124
Nama : Ila
Kelas : 2A
IPK : 3.1
-----
NIM : 127
Nama : Udin
Kelas : 2A
IPK : 3.2
-----
NIM : 126
Nama : Tika
Kelas : 2A
IPK : 3.3
-----
NIM : 125
Nama : Agus
Kelas : 2A
IPK : 3.6
-----
NIM : 123
Nama : Ali
Kelas : 2A
IPK : 3.9
-----
```

5.3.5 Jawaban Pertanyaan

Proses tersebut digunnakan untuk mencari elemen dengan nilai ipk terkeciil dalam subarray yang belum diurutkan

5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

Verifikasi Hasil program

```
Data Mahasiswa Sebelum sorting:
NIM : 123
Nama : Ayu
Kelas : 2A
IPK : 3.7
-----
NIM : 124
Nama : dika
Kelas : 2A
IPK : 3.0
-----
NIM : 125
Nama : ila
Kelas : 2A
IPK : 3.8
-----
NIM : 126
Nama : susi
Kelas : 2A
IPK : 3.1
-----
NIM : 127
Nama : yayuk
Kelas : 2A
IPK : 3.4
-----
Data yang sudah teurut menggunakan INSERTION SORT (ASC) :
NIM : 124
Nama : dika
Kelas : 2A
IPK : 3.0
-----
NIM : 126
Nama : susi
Kelas : 2A
IPK : 3.1
-----
NIM : 127
Nama : yayuk
Kelas : 2A
IPK : 3.4
-----
NIM : 123
Nama : Ayu
Kelas : 2A
IPK : 3.7
-----
NIM : 125
Nama : ila
Kelas : 2A
IPK : 3.8
-----
```

5.4.1 Jawaban Pertanyaan

```
void selectionSort() {
    for (int i = 0; i < listMhs.length - 1; i++) {
        int idxmin = i;
        for (int j = i + 1; j < listMhs.length; j++) {
            if (listMhs[j].ipk < listMhs[idxmin].ipk) {
                idxmin = j;
            }
        }
        Mahasiswa14 temp = listMhs[idxmin];
        listMhs[idxmin] = listMhs[i];
        listMhs[i] = temp;
    }
}
```

5.5 Latihan Praktikum

```
package Praktikum05;

import java.util.Scanner;

class Dosen14 {
    String kode;
    String nama;
    String jenisKelamin;
    int usia;

    Dosen14(String kode, String nama, String jenisKelamin, int usia) {
        this.kode = kode;
        this.nama = nama;
        this.jenisKelamin = jenisKelamin;
        this.usia = usia;
    }

    void tampil() {
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis Kelamin: " + jenisKelamin);
        System.out.println("Usia: " + usia);
    }
}

class DataDosen {
    Dosen14[] dataDosen = new Dosen14[10];
    int idx = 0;

    void tambah(Dosen14 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }
}
```



```
}
```

```
void tampil() {  
    for (int i = 0; i < idx; i++) {  
        dataDosen[i].tampil();  
        System.out.println("-----");  
    }  
}
```

```
void sortingASC() {  
    for (int i = 0; i < idx - 1; i++) {  
        for (int j = 1; j < idx - i; j++) {  
            if (dataDosen[j].usia < dataDosen[j - 1].usia) {  
                Dosen14 temp = dataDosen[j];  
                dataDosen[j] = dataDosen[j - 1];  
                dataDosen[j - 1] = temp;  
            }  
        }  
    }  
}
```

```
void sortingDSC() {  
    for (int i = 0; i < idx - 1; i++) {  
        int idxMax = i;  
        for (int j = i + 1; j < idx; j++) {  
            if (dataDosen[j].usia > dataDosen[idxMax].usia) {  
                idxMax = j;  
            }  
        }  
        Dosen14 temp = dataDosen[idxMax];  
        dataDosen[idxMax] = dataDosen[i];  
        dataDosen[i] = temp;  
    }  
}
```

```
void insertionSortDSC() {  
    for (int i = 1; i < idx; i++) {
```

```

        Dosen14 temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}

}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DataDosen dataDosen = new DataDosen();
        int pilihan;

        do {
            System.out.println("Menu:");
            System.out.println("1. Tambah data dosen");
            System.out.println("2. Tampil data dosen");
            System.out.println("3. Sorting ASC (usia termuda ke tertua)");
            System.out.println("4. Sorting DSC (usia tertua ke termuda)");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Kode: ");
                    String kode = scanner.nextLine();
                    System.out.print("Nama: ");
                    String nama = scanner.nextLine();
                    System.out.print("Jenis Kelamin: ");
                    String jenisKelamin = scanner.nextLine();
                    System.out.print("Usia: ");

```

```

        int usia = scanner.nextInt();

        scanner.nextLine();

        Dosen14 dsn = new Dosen14(kode, nama, jenisKelamin, usia);
        dataDosen.tambah(dsn);

        break;
case 2:
        dataDosen.tampil();

        break;
case 3:
        dataDosen.sortingASC();

        System.out.println("Data dosen setelah diurutkan (ASC):");
        dataDosen.tampil();

        break;
case 4:
        System.out.println("Pilih metode sorting:");
        System.out.println("1. Selection Sort");
        System.out.println("2. Insertion Sort");
        int metode = scanner.nextInt();
        scanner.nextLine();

        if (metode == 1) {
            dataDosen.sortingDSC();
        } else if (metode == 2) {
            dataDosen.insertionSortDSC();
        } else {
            System.out.println("Metode tidak valid");
        }

        System.out.println("Data dosen setelah diurutkan (DSC):");
        dataDosen.tampil();

        break;
case 5:
        System.out.println("Keluar dari program.");

        break;
default:
        System.out.println("Pilihan tidak valid.");

```

```
        }  
    } while (pilihan != 5);  
  
}  
  
}
```

5.5.1 Link Git Hub

<https://github.com/ianmen10/SEMESTER-genap2.git>